****

数据库原理

实 验 报 告

|  |  |
| --- | --- |
| **学 院** | 网络空间安全学院 |
| **专 业** | 网络工程 |
| **班 级** | 18272412 |
| **学 号** | 18041618 |
| **学生姓名** | 廖越强 |
| **教师姓名** | 丁宏 |
| **完成日期** | 11.23 |
| **成 绩** |  |
| **实验一 利用SSMS创建与管理数据库** | | |
| 1. **实验目的**   T-SQL 语句的熟练应用，对 SELECT 语句进行反复应用与操作。 | | |
| 1. **实验内容**   （1）利用 T-SQL 数据库的创建与维护  （2）利用 T-SQL 表的建立与维护  （3）SELECT 语句的应用 | | |
| 1. **实验环境**   Windows10 64位系统  SQL Server 2008 Express | | |
| 1. **主要操作步骤及实验结果记录**   **任务一：利用 T-SQL 数据库的创建与维护**   1. 创建创建“学生信息”数据库“stuinfo1”   create database stuinfo1 on primary (  Name = stuinfo1,  Filename = 'E:\database\stuinfo1.mdf',  Size = 5MB,  Maxsize = unlimited,  Filegrowth = 10%  )  log on  (  Name = stu1\_log,  Filename = 'E:\database\stu1.ldf',  Size = 3MB,  Maxsize = 50MB,  Filegrowth = 2MB  )  2.删除数据库  DROP DATABASE stuinfo1  **任务二：表的创建与维护**   1. 定义表（基本表/用户表）结构   CREATE TABLE tab\_student(  sno char(4) NOT NULL PRIMARY KEY,  sname varchar(50) NULL,  ssex char(2) NULL CHECK(ssex='男' or ssex='女'),  birthday datetime NULL,  age AS YEAR(GETDATE())-YEAR(birthday),  class varchar(10) NULL,  grade tinyint NULL,  majorno char(4) NULL REFERENCES tab\_major(majorno))  2.修改表数据  添加数据  INSERT INTO tab\_student (sno,sname,ssex,birthday,class,grade )  VALUES ('0101'，'李一', ‘男' ， ‘1999-1-1' ， ‘2018’)  更新数据  UPDATE tab\_score SET score=90 WHERE cno=’0101’  新增字段 alter table tab\_institute add dean varchar(10) null  其余的表也是类似  **任务三：SELECT 语句应用**   1. 单表查询 2. 查询 t tab\_student 表所有学生的记录   SELECT \* FROM tab\_student     1. 查询tab\_student 表中前 4 4 个学生的记录   SELECT TOP 4 \* FROM tab\_student     1. 查询 tab\_student 表所有学生的学号和姓名   SELECT sno,sname FROM tab\_student     1. 查询tab\_student 表专业号为“ 0101”   SELECT \* FROM tab\_student WHERE majorno=0101     1. 查询tab\_student 表专业号为“ 0101”。结果按年龄从小到大排序   SELECT \* FROM tab\_student WHERE majoyno=0101 ORDER BY Birthday DESC     1. 查询tab\_student表专业号为“0101”的学生的学号，姓名和年龄，并将列名改成中文**。**   SELECT sno as "学号",sname as "姓名",YEAR(GETDATE())-YEAR(Birthday) as "年龄" FROM tab\_student  WHERE majoyno=0101 ORDER BY Birthday DESC     1. 查询tab\_couse表中含有“数据”的课程记录。（查询条件中LIKE与通配符的用法）   SELECT \* FROM tab\_course WHERE cname LIKE '%数据%'     1. a查询tab\_student表专业号为“0101”、“0102”的学生记录（查询条件中IN的用法）   SELECT \* FROM tab\_student WHERE majoyno IN (0101,0102)     1. a查询tab\_score表中成绩为空的记录（查询条件中IS NULL的用法）   SELECT \* FROM tab\_score WHERE score is NULL     1. a查询tab\_score表中成绩为80-89的记录（查询条件中BETWEEN的用法）   SELECT \* FROM tab\_score WHERE score BETWEEN 80 AND 90     1. 多表查询 2. 查询全校的教师的工号、姓名、所在学院名   SELECT tno,tname,instname FROM tab\_teacher a,tab\_institute b WHERE a.instno=b.instno  SELECT tno,tname,instname FROM tab\_teacher a LEFT JOIN tab\_institute b ON a.instno=b.instno       1. 查询授课教师的工号、姓名、讲授课程名   SELECT t.tno,tname,cname FROM tab\_teacher t , tab\_course c, tab\_score s WHERE t.tno=s.tno AND s.cno=c.cno  SELECT t.tno,tname,cname FROM tab\_teacher t INNER JOIN tab\_score s ON t.tno=s.tno INNER JOIN tab\_course c ON s.cno=c.cno       1. 查询 “数据结构”课程的学生的学号、姓名、课程名、成绩、教师名   SELECT s.sno,sname,cname,score,tname FROM tab\_score sc,tab\_student s,tab\_teacher t,tab\_course c WHERE s.sno=sc.sno AND t.tno=sc.tno AND cname='数据结构'  SELECT s.sno,sname,cname,score,tname FROM tab\_score sc INNER JOIN tab\_student s ON s.sno=sc.sno INNER JOIN tab\_course c ON cname='数据结构' INNER JOIN tab\_teacher t ON t.tno = sc.tno       1. 查询“计算机学院”所有学生的学号、姓名、课程名、成绩、教师名   SELECT Stu.sno, sname, cname, score, tname FROM tab\_student Stu, tab\_score Sc, tab\_teacher T, tab\_course Cou, tab\_major M, tab\_institute Ins WHERE Stu.sno=Sc.sno AND Sc.tno=T.tno AND Sc.cno=Cou.cno AND Stu.majoyno=M.majoyno AND M.instno=Ins.instno AND Ins.instname='计算机学院'  SELECT s.sno, sname, cname, score, tname  FROM tab\_student s  inner join tab\_institute i on i.instname='计算机学院'  inner join tab\_major m on m.majoyno=s.majoyno  inner join tab\_score sc on s.sno=sc.sno  inner join tab\_teacher t on t.tno = sc.tno and t.instno=i.instno  inner join tab\_course c on c.cno=sc.cno       1. 查询所有授课的“教授”的工号、姓名和授课课程名   SELECT t.tno,tname,cname FROM tab\_teacher t,tab\_course co,tab\_score s WHERE t.tno = s.tno AND s.cno=co.cno AND t.title='教授'  SELECT t.tno,tname,cname FROM tab\_teacher t INNER JOIN tab\_course c ON t.title='教授' INNER JOIN tab\_score s ON t.tno = s.tno AND c.cno = s.cno       1. 嵌套查询的操作练习 2. 查询“计算机学院”的教师记录   SELECT \* FROM tab\_teacher where instno IN (SELECT instno FROM tab\_institute WHERE instname='计算机学院')     1. 查询“计算机学院”的课程记录   SELECT \* FROM tab\_course where majorno IN (SELECT majorno FROM tab\_major WHERE instno IN (SELECT instno FROM tab\_institute WHERE instname='计算机学院'))     1. 查询讲授“数据库”课程的教师记录   SELECT \* FROM tab\_teacher WHERE tno IN (SELECT tno FROM tab\_score WHERE cno= (SELECT cno FROM tab\_course WHERE cname='数据库'))     1. 查询选修“数据结构”课程的学生记录   SELECT \* FROM tab\_student WHERE sno IN (SELECT sno FROM tab\_score WHERE cno IN (SELECT cno FROM tab\_course WHERE cname = '数据结构'))     1. 查询“程序设计”课程成绩第一名的学生的学号与姓名   SELECT TOP 1 sno,sname FROM tab\_student WHERE sno IN (SELECT sno FROM tab\_score WHERE cno = (SELECT cno FROM tab\_course WHERE cname = '课程设计'))     1. 查询“程序设计”课程成绩高于“李四”的学生的学号与成绩   SELECT sno, score FROM tab\_score WHERE cno=(SELECT cno FROM tab\_course WHERE cname='课程设计')AND score <(SELECT score FROM tab\_score WHERE cno=(SELECT cno FROM tab\_course WHERE cname='课程设计') AND sno= (SELECT sno FROM tab\_student WHERE sname='李四'))     1. 查询所有授课的“教授”的工号、姓名和授课课程   SELECT tno,tname,cname FROM tab\_teacher t,tab\_course c WHERE t.titile='教授' AND t.tno IN (SELECT tno FROM tab\_score s WHERE cno is not null AND c.cno = s.cno)     1. 查询选修了“计算机科学与技术”专业所有课程的学生学号和姓名。（用NOT EXISTS实现）   SELECT sno, sname FROM tab\_student WHERE NOT EXISTS (SELECT \* FROM tab\_course WHERE NOT EXISTS (SELECT \* FROM tab\_score WHERE  tab\_student.sno=sno AND tab\_course.cno='计算机科学与技术'))    （四）统计查询  统计查询   * 1. 统计计算机学院的教师人数   2. 统计网络工程专业每个学生的选课课程数   3. 统计“数据库”课程的最高分、最低分和平均成绩   4. 查询“数据库”课程高于平均分的学生学号和成绩   5. 查询“数据库”课程成绩最高的学生学号和姓名   6. 查询平均分高于75分的课程号   7. 查询计算机学院平均分高于75分的学生学号   8. 统计有不及格成绩的学生人数   9. 查询选课人数少于10人的课程号   10. 查询所有学生的学号、姓名和所选课程的最高分、最低分、平均分(用派生表实现)  1. select COUNT(\*) from tab\_teacher      1. SELECT COUNT(\*) AS n FROM tab\_student WHERE majoyno= ( SELECT majoyno FROM tab\_major WHERE majoyname='网络工程' )      1. SELECT MAX(score) AS 最高分, MIN(score) AS 最低分, AVG(score) AS 平均分 FROM tab\_score WHERE cno= ( SELECT cno FROM tab\_course WHERE cname='数据库')      1. select sno, score from tab\_score where score > ( select AVG(score) as '平均分' from tab\_score where cno= ( select cno from tab\_course where cname='数据库' ) )      1. select sno, sname from tab\_student where sno in ( select sno from tab\_score where score >= ( select MAX(score) as 'max' from tab\_score ) )      1. select cno from tab\_score group by cno having AVG(score) > 75      1. select s.sno from tab\_student s, tab\_major m, tab\_institute i, tab\_score sc where s.sno=sc.sno and s.majoyno=m.majoyno and m.instno=i.instno and i.instname='计算机学院' group by s.sno having AVG(score)>75      1. select COUNT(\*) as '不及格' from tab\_score where score < 60 group by sno      1. select c.cno from tab\_course c, tab\_student s, tab\_score sc where s.sno=sc.sno and sc.cno=c.cno group by c.cno having COUNT(\*)<10      1. select sc.sno, sname, max\_score, min\_score, avg\_score from tab\_score sc, tab\_student s, ( select sno, MAX(score), MIN(score), AVG(score) from tab\_score Group by sno ) as c(sno, max\_score, min\_score, avg\_score) where sc.sno=c.sno and sc.sno=s.sno | | |
| 1. **实验分析总结及心得**   本次实验任务量太过大。已经超出了我其他所有课程加起来的的作业量了，希望以后能调整。另外涉及到的课程知识也相当地多，我认为这需要更多的时间才能够掌握。很多复杂的语句虽然参照ppt写了出来，但我认为我仍然是没有完全掌握的完整的设计方法，需要很长一段时间来消化吧。  通过这次实验，我对select的操作熟悉了许多，因为时间基本上都花在了选择查询语句上，简单、中等难度的已经基本掌握，但非常复杂的那些还需要一段时间来强化。印象比较深的是除了可以使用WHERE子句表达连接条件外，还可以通过from子句的方式来进行多表连接，这两种方式都挺好用的，各有优点吧。另外就是嵌套查询，这个真的很容易绕晕，摸索好一段时间后，才渐渐摸到门道----要从最基本的子句开始构思，也就是把最里面的子句先写好，然后看看最终需要选取什么信息，之后再从里到外慢慢构思，一层一层往外直到得到我们想要的信息。总的来说，还是学会了不少知识和小技巧的。 | | |